École Polytechnique Fédérale de Lausanne
Semestre d'automne 2020
Prof. Laurent Villard, Dr Cristian Sommariva

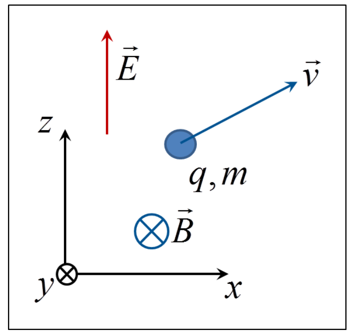
Physique Numérique I – Exercice 2

A rendre jusqu'au mercredi 28 octobre 2020 sur le site
http://moodle.epfl.ch/mod/assignment/view.php?id=835968

2 Particule chargée dans un champ électromagnétique

Un particule de masse $m = 1.6726 \times 10^{-27}$kg, de charge $q = 1.6022 \times 10^{-19}$C, avec une position initiale (x_0, z_0) et une vitesse initiale $\vec{v}_0 = (v_{x0}, v_{z0})$, est plongé dans un champ électrique uniforme $\vec{E} = E_0\hat{x}$ et un champ magnétique uniforme $\vec{B} = B_0\hat{y}$. Il est alors soumis à la force de Lorentz

$$\vec{F} = q(\vec{E} + \vec{v} \times \vec{B}). \tag{1}$$



Le but de cet exercice, du point de vue physique, est d'étudier les trajectoires de la particule. Pour simplifier, on examinera le cas de trajectoires dans le plan (x, z).

Du point de vue numérique, le but est d'étudier les propriétés de stabilité et de convergence de cinq schémas différents : Euler explicite, Euler implicite, Euler-Cromer, Runge-Kutta d'ordre 2 et Boris-Buneman.

- Le schéma d'Euler explicite est décrit par l'Eq.(2.11) de Notes de Cours, avec $\mathbf{y} = (x, z, v_x, v_z)$.
- Le schéma d'Euler implicite est décrit à la section 2.3.4 des Notes de Cours.
- Le schéma d'Euler-Cromer est, avec la position $\vec{x} = (x, z)$, la vitesse $\vec{v} = (v_x, v_z)$ et l'accélération $\vec{a} = (a_x, a_z)$:
  - $\vec{x}_{n+1} = \vec{x}_n + \vec{v}_{x,n}\Delta t$
  - $v_{x,n+1} = v_{x,n} + a_x(v_{z,n})\Delta t$
  - $v_{z,n+1} = v_{z,n} + a_z(v_{x,n+1})\Delta t$
- Le schéma de Runge-Kutta d'ordre 2 est décrit par l'Eq.(2.97) des Notes de Cours.
- Le schéma de Boris-Buneman s'écrit, en posant $\omega_c = qB/m$, $\beta = \omega_c\Delta t$:
  - $\vec{x}_- = \vec{x}_n + \vec{v}_n\Delta t/2$
  - $\vec{v}_- = \vec{v}_n + (q/m)\vec{E}\Delta t/2$
  - $v_{x,+} = \left[\left(1 - \beta^2/4\right) v_{x,-} - \beta v_{z,-}\right] / \left(1 + \beta^2/4\right)$
  - $v_{z,+} = \left[\left(1 - \beta^2/4\right) v_{z,-} + \beta v_{x,-}\right] / \left(1 + \beta^2/4\right)$
  - $\vec{v}_{n+1} = \vec{v}_+ + (q/m)\vec{E}\Delta t/2$
  - $\vec{x}_{n+1} = \vec{x}_- + \vec{v}_{n+1}\Delta t/2$

2.1 Calculs analytiques [10pts]

(a) **[3 pts]** Obtenir le système d'équations différentielles du mouvement et l'écrire sous la forme $d\mathbf{y}/dt = \mathbf{f}(\mathbf{y})$ pour $\mathbf{y} = (x, z, v_x, v_z)$.

(b) **[3 pts]** Ecrire l'expression de l'énergie mécanique et montrer qu'elle est conservée.

(c) **[4 pts]** Tourver la solution analytique pour la condition initiale suivante : $\vec{x}(0) = 0$, $\vec{v}(0) = v_0\hat{z}$. *Indication : Dans un premier temps, considérer le cas $\vec{E} = 0$, $\vec{B} \neq 0$ et résoudre le système* *pour (v_x, v_z) en faisant l'analogie avec le mouvement oscillatoire harmonique. Puis, dans un deuxième temps, rajouter le champ électrique, $\vec{E} \neq 0$, $\vec{B} \neq 0$.*

2.2 Implémentation en C++

Télécharger le fichier Exercice2.zip du site Moodle. Nous utiliseront un vecteur de type 'valarray', qui est implémenté dans la 'Standard Template Library' de C++. Voir la documentation sous www.cplusplus.com/reference/valarray/valarray/. Dans le code, il faut en particulier implémenter les schémas d'Euler explicite, Euler implicite, Euler-Cromer, Runge-Kutta d'ordre 2 et Boris-Buneman. Il faudra aussi implémenter le calcul de l'énergie mécanique et du moment magnétique définie au point 2.3.c.

2.3 Simulations et Analyses

On effectue des simulations avec le programme que l'on vient d'écrire et de compiler. Si l'éxécutable est `Exercice2`, taper à la ligne de commande d'un Terminal :

```
./Exercice2  configuration.in
```

La visualisation des résultats numériques se fait avec `Matlab` ou Python. Voir les fichiers Matlab : Analyse.m et ParameterScan.m ou Python : Analyse.py, PlotResults.py et SchemeAnalysisSuite.py, que vous modifierez selon vos besoins.

(a) **[15 pts] Cas sans champ électrique mais avec champ magnétique :**
Considérer le cas $E_0 = 0$, $B_0 = 5$T et la condition initiale $\vec{x}(0) = 0$, $\vec{v}(0) = v_0\vec{e}_z$, $v_0 = 4\times10^5$m/s. On prendra t_{fin} tel que le proton effectue théoriquement 5 périodes de rotation.

   (i) **[9 pts]** Effectuer des simulations avec des nombres de pas de temps (nsteps) différents. A chaque pas de temps j, calculer l'erreur locale $e_j = |\,(f(t_j) - f_j)\,|$ entre chaque composante de la position et de la vitesse de la solution numérique (f_j) et celles de la solution exacte ($f(t_j)$) calculée en2.1 (c). Pour chaque simulation, obtenir le maximum de cette erreur locale, $\epsilon = \max_j e_j$, que l'on définira comme étant l'erreur numérique. Vérifier la convergence numérique de l'erreur ϵ et déterminer l'ordre de convergence, pour les cinq schémas numériques implémentés. *Indication : vous pouvez utiliser le script Matlab* `ParameterScan.m`*, en l'adaptant, pour automatiser le lancement d'une série de simulations ou le script Python* `Analyse.py`*.*

   (ii) **[6 pts]** Tracer la non-conservation de l'énergie mécanique, $\Delta E_{\text{mec}}(t) = E_{\text{mec}}(t) - E_{\text{mec}}(0)$, en fonction du temps. Montrer (illuster) que les schémas d'Euler explicite et de Runge-Kutta d'ordre 2 sont instables (i.e. l'erreur croît exponentiellement dans le temps), alors que Euler-Cromer est stable pour autant que $\omega_c\Delta t < 2$, Euler implicite et Boris-Buneman sont stables por toute valeur de Δt. Montrer (illustrer) que le schéma de Boris-Buneman conserve exactement l'énergie, aux erreurs d'arrondi près, alors que pour les autres schémas la conservation n'est pas exacte.

(b) **[10 pts] Cas avec champ électrique et champ magnétique : dérive $\vec{E} \times \vec{B}$**
On prend $E_0 = 10^5$V/m, $B_0 = 5$T, $\vec{x}(0) = 0$, $v_{x0} = 0$, $v_{z0} = 4 \cdot 10^5$m/s, et le même t_{fin} qu'en 2.3(a). Effectuer des simulations avec le schéma de Runge-Kutta d'ordre 2 et le schéma de Boris-Buneman.

   (i) **[3pts]** Faire une étude de convergence en Δt.

   (ii) **[3pts]** Vérifier la conservation de l'énergie.

   (iii) **[4pts]** Représenter la trajectoire de la particule dans un référentiel en translation uniforme à la vitesse $v_E = \vec{E} \times \vec{B}/B^2$.

(c) **[10 pts] Cas d'un champ magnétique non uniforme : dérive $\vec{B} \times \nabla B$**
On considère maintenant un champ magnétique d'intensité variable $\vec{B} = B(x)\hat{y}$, avec $B(x) = B_0(1 + \kappa x)$, $\kappa = 100\text{m}^{-1}$, $B_0 = 5\text{T}$, $E = 0$, $v_{x0} = 0$, $v_{z0} = 4 \cdot 10^5\text{m/s}$, $x_0 = mv_{z0}/qB_0$, $z_0 = 0$ et un t_{fin} dix fois plus grand que celui du point 2.3(a) (environ 50 périodes de rotation). Effectuer des simulations avec les schémas de Runge-Kutta d'ordre 2 et de Boris-Buneman.

  (i) **[5pts]** Observer ce qui se passe avec un proton ($q = e$) et un antiproton ($q = -e$). *Indication : attention, le signe de la charge q intervient dans la définition de x_0.*

  (ii) **[5pts]** Calculer et étudier la quantité $\mu = mv^2/2B$. *N.B. : C'est le moment magnétique d'une particule libre dans un champ magnétique. On peut montrer, sous certaines hypothèses, que c'est un invariant adiabatique, i.e. approximativement constante du mouvement.*

(d) **[5 pts]** *Facultatif. Le but de cette section est de stimuler votre créativité. N'hésitez pas à vous lancer si vous avez d'autres idées.*

  (i) Généraliser le code à des trajectoires 3D et des champs magnétiques $\vec{B}(\vec{x})$ 3D.

  (ii) Choisir d'autres champs magnétiques $\vec{B}(\vec{x})$, par exemple avec courbure, et étudier les trajectoires obtenues. (Exemple : particules du vent solaire dans le champ magnétique terrestre). (N.B. : le champ magnétique doit enprincipe satisfaire la condition $\nabla \cdot \vec{B} = 0$, ce qui n'est généralement pas un problème trivial, sauf s'il y a des symétries dans le problème. Le cas général 3D va bien au-delà d'un cours de physique générale de 2e année).

N.B. On trouve plusieurs documents `Matlab` (introduction, examples, références) dans un dossier spécifique sur notre site "Moodle" (Dossier Matlab).

2.4 Rédaction du rapport en LaTeX

(a) Télécharger le fichier source en `tex` (SqueletteRapport.tex) sur "Moodle", ou partir du rapport précédent.

(b) Rédiger un rapport dans lequel les résultats des simulations ainsi que les réponses aux questions ci-dessous sont présentées en détail.

N.B. On trouve plusieurs documents LaTeX (introduction, examples, références, problème des accents avec Kile) dans un dossier spécifique sur Moodle (Dossier LaTeX).

2.5 Soumission du rapport et du code C++

(a) Préparer le fichier du rapport en format `pdf` portant le nom `RapportExercice2_Nom1_Nom2.pdf`,

(b) Préparer le fichier source C++ `Exercice2_Nom1_Nom2.cpp`.

(c) Le lien de soumission se trouve dans la section du premier exercice sur Moodle et peut directement être atteint en cliquant ici.

N.B. : En plus des points mentionnés ci-dessus, **[5 pts]** sont attribués pour la qualité générale de la présentation du rapport (clarté des arguments, des figures, etc.)